Översikt

- Använda AJAX och Partial Page Updates.
- Implementera strategi för mellanlagring.

.eo/ee

Lektion 1: Använda AJAX och Partial Page Updates

- Introduktion.
- Varför använda Partial Updates?
- Använda AJAX.
- AJAX -exempel.
- Ajax.ActionLink Helper.
- Ajax.ActionLink Helper -exempel.

.eo/ee

Introduktion

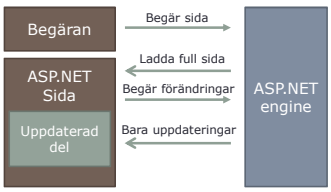
- Många webbapplikationer behöver visa stora mängder information och oftast även mycket grafik.
- Stor volym av information, för att sidan laddas långsamt.
- Partial page, innebär att bara delar av sidan kommer att laddas in.

Varför använda Partial Updates?

- ASP.NET och MVC arbetar med serverbaserad bearbetning. Server genererar HTML-sidan.
- När sidan uppdateras, kommer server att kontaktas och ny sida kommer att genereras.
- Detta påverkar prestandan för din applikation.
- Utvecklingsmodellen AJAX reducerar behovet av återinläsning av hela sidan.
- AJAX använder sig av JavaScript och XML för att ta reda på information från klientsystemet, baserat på denna information skapar AJAX webbsidan som sedan skickas från server.

Varför använda Partial Updates? (forts.)

- Partial Page Updates använder sig av AJAX för att uppdatera individuella delar av sidan under POSTBACK, istället för hela sidan.



Använda AJAX

- För att implementera AJAX i din MVC-applikation, behöver du skapa View som renderar bara det uppdaterade innehållet, inte hela sidan.
- Börja med att utveckla din applikation, utan AJAX, när applikationen fungerar som det är tänkt, titta på var du kan förbättra prestandan.
- För att implementera partial page updates, behöver du skapa View som bara innehåller sektion som du vill uppdatera.

Använda AJAX (forts.)

- I klassen ViewController, behöver du uppdatera funktionen View med att returnera klassen PartialView, istället för hela klassen View.
- Som tillägg kan du lägga till attributen HttpGet eller HttpPost, innan funktionen för View.
- Attributen visar om partial page updates skall ske över metoden HTTPPost eller HTTPGet.

Använda AJAX (forts.)

- Med AJAX, kommer JavaScript att hämta bara en specifik del av sidan, den del som du vill uppdatera, från servern.

AJAX -exempel

```
public class DefaultController : Controller
{
        //
        // GET: /Default1/
        public ActionResult Index()
        {
                ViewBag.Message = ”Hello”;
                return View();
        }
        [HttpGet]
        public PartialViewResult HelloWorld()
        {
                ViewBag.Message = ”Hello World!”;
                return PartialView();
        }
}
```

.eo/ee

Ajax.ActionLink Helper

- Ajax.ActionLink helper kan användas för att trigga partial page updates.
- Helper hjälper till att initiera JavaScript för att få den uppdaterade HTML-informationen från View och ersätta eller lägga till den uppdaterade information på specifik plats.

.eo/ee

Ajax.ActionLink Helper -exempel

```
@{
        Layout = null;
<!DOCTYPE html>
<html>
<head>
        <meta name=”viewport” content=”width=device-width” />
        <title>Index<title>
        <script src=”http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.8.3.js”
type=”text/javascript”></script>
        <script
src=http://ajax.aspnetcdn.com/ajax/mvc/3.0/jquery.unobstrutive-ajax.min.js
type=”text/javascript”></script>
</head>
<body>
        <div>
           <div id=”divMessage”>@ViewBag.Message</div>
                @AjaxActionLink(”Refresh”, ”HelloWorld”, new AjaxOptions{
HttpMethod = ”POST”, UpdateTargetId = ”divMessage”, InsertionMode =
InsertionMode.Replace })
        </div>
</body>
</html>
```

.eo/ee

Lektion 2: Implementera strategi för mellanlagring

- Varför använda mellanlagring?
- Output cache.
- Data cache.
- Data cache -arbetsflöde.
- HTTP cache.
- Förhindra mellanlagring.
- Förhindra mellanlagring –exempel.

Varför använda mellanlagring?

- När teknik för mellanlagring används, involverar detta att mellanlagra information som har hämtats från databab i minnet på webbserver.
- Om det är statisk information, kan innehållet lagras i cache eller på proxy server.
- Om information används från cache, eliminerar detta behovet av bearbetning.
- Tekniken hjälper ***inte*** webbapplikation som inkluderar innehåll som uppdateras frekvent.

Output cache

- Output cache ger möjlighet för ASP.NET engine att lagra renderat innehåll av webbsida i webbserverns minne.
- Funktionen är ett bra komplement till AJAX partial page update.
- Lägg till attributet Output cache i controller.

Data cache

- Webbapplikationer är oftast beroende av innehål från databas, för rendering av innehåller för en webbsida.
- Databaser kan ibland råka ut för bekymmer med prestanda.
- Data cache kan implementeras i applikationen för att förhindra laddning av information från databas, varje gång användare skickar en begäran.

.eosec

Data cache -arbetsflöde

1. Laddar information från databas.
2. Lagrar innehåll i Memory cache.
3. Hämta innehåll från Memory cache.
4. Försäkra dig om att det finns information, annars återinladda innehåll.

.eosec

HTTP cache

- De flesta webbläsare lagrar innehåll som är nerladdat från webbserver, i lokal cache.
- Eliminerar behovet av återinladdning.
- Kontrollera om innehållet har förändrats, om inte använd information från cache.

.eosec

Förhindra mellanlagring

- Att arbeta med mellanlagring kan ibland bli bekymmer för webbapplikationen.
- Framförallt om frekventa uppdateringar används, cache kan förhindra att dessa uppdatering kan ses.
- För att förhindra detta, kan HTTP header Cache-Control implementeras.
- Alla HTTP-klienter, webbläsare eller proxy server svarar på de instruktioner som finns i denna header, för att hantera den lokala mellanlagringen.

.easec

Förhindra mellanlagring -exempel

```
Response.Cache.SetCacheability(HttpCacheability.NoCache);
```

.easec

Övning Konfigurera mellanlagring

.easec

Repetitionsfrågor